This Page Is Inserted by IFW Operations
and is not a part of the Official Record

# BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

## IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images, please do not report the images to the Image Problem Mailbox.**

(12) UK Patent Application (19) GB (11) 2 350 990 (13) A

(43) Date of A Publication 13.12.2000

(21) Application No 9925012.8

(22) Date of Filing 25.10.1999

(30) Priority Data
(31) 09189261 (32) 10.11.1998 (33) US

(51) INT CL[7]
G06F 9/44

(52) UK CL (Edition R )
H4T TDXX

(56) Documents Cited
WO 99/17193 A1 WO 98/06033 A1
"XML Update:An XML GUI Programming Tool",
www.webtools.com/...intableArticle?doc_id=
TLS19980805S0008

(58) Field of Search
UK CL (Edition R ) H4T TDXX
INT CL[7] G06F 9/44 9/45
Online: WPI, INSPEC, EPODOC, JAPIO, TDB,
COMPUTER, Internet

(71) Applicant(s)
International Business Machines Corporation
(Incorporated in USA - New York)
Armonk, New York 10504, United States of America

(72) Inventor(s)
Charles Burkettinski
David Bruce Lection

(74) Agent and/or Address for Service
C J Ling
IBM United Kingdom Limited, Hursley Park,
WINCHESTER, Hampshire, SO21 2JN,
United Kingdom

(54) Abstract Title
**Initializing graphic user interface controls with extensible markup language-formatted data**

(57) Systems, methods and computer program products are provided for initializing, via a computer operating system, graphical user interface (GUI) controls for an application program running within the computer operating system GUI. An Extensible Markup Language (XML) document associated with the GUI control is retrieved via initialization code stored within the computer operating system. The XML document includes at least one string of text that indicates how the GUI control is to be displayed within the GUI. The retrieved XML document is parsed by the computer operating system to identify each string of text contained therewithin. The GUI control is then initialized within the GUI using each identified string of text.

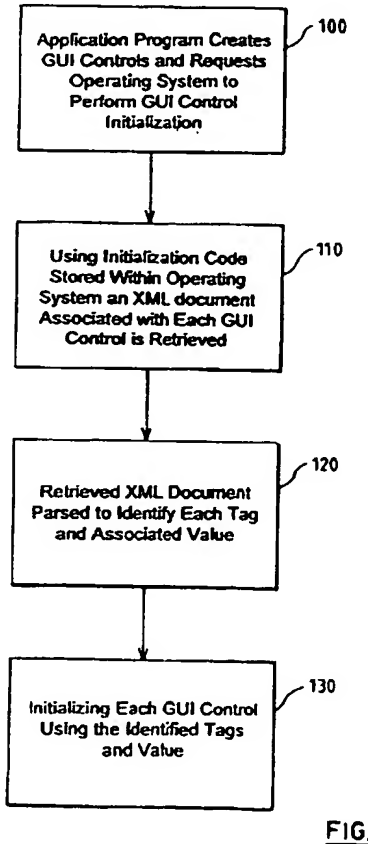


FIG. 6

GB 2 350 990 A

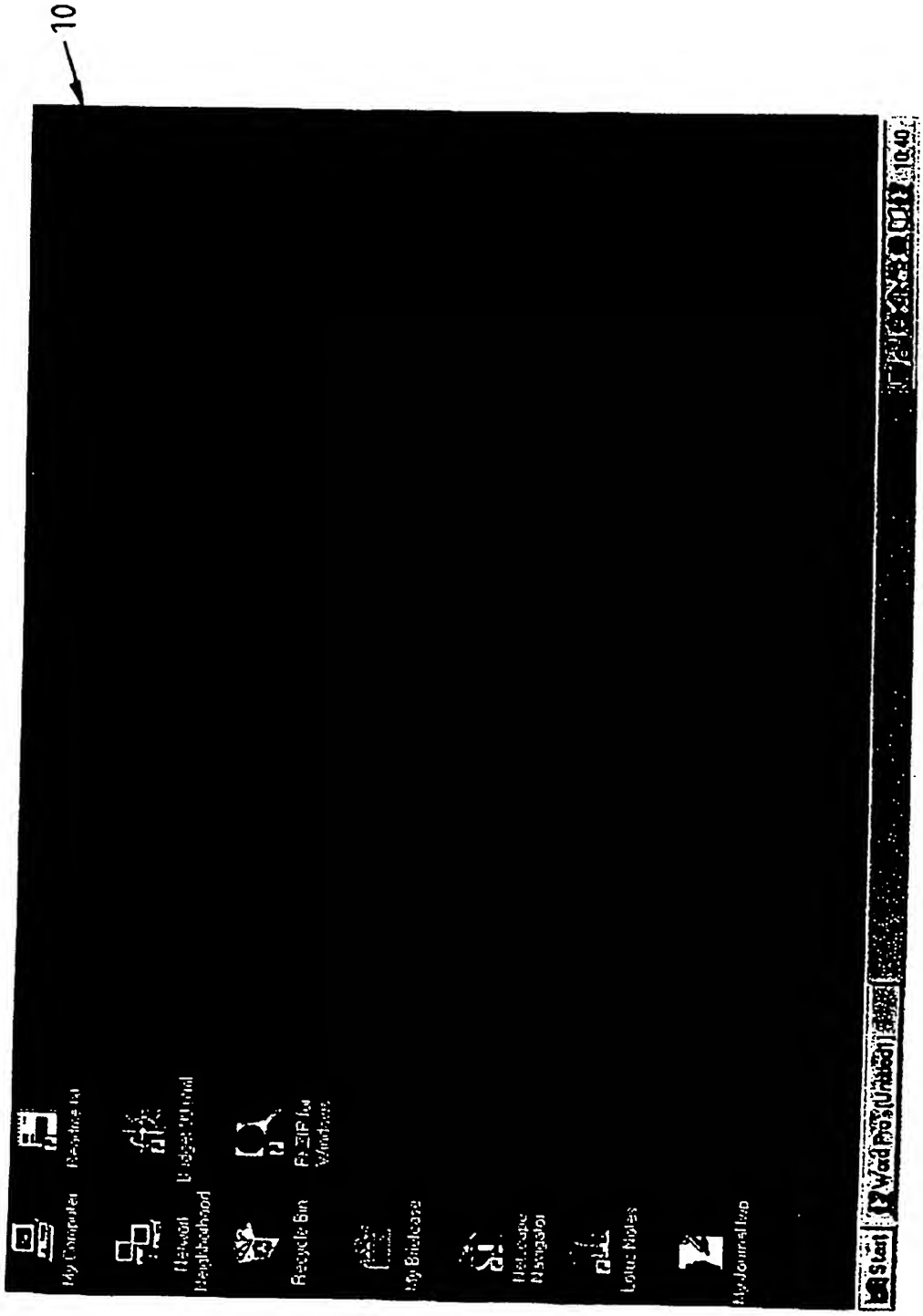


FIG. 1 (PRIOR ART)

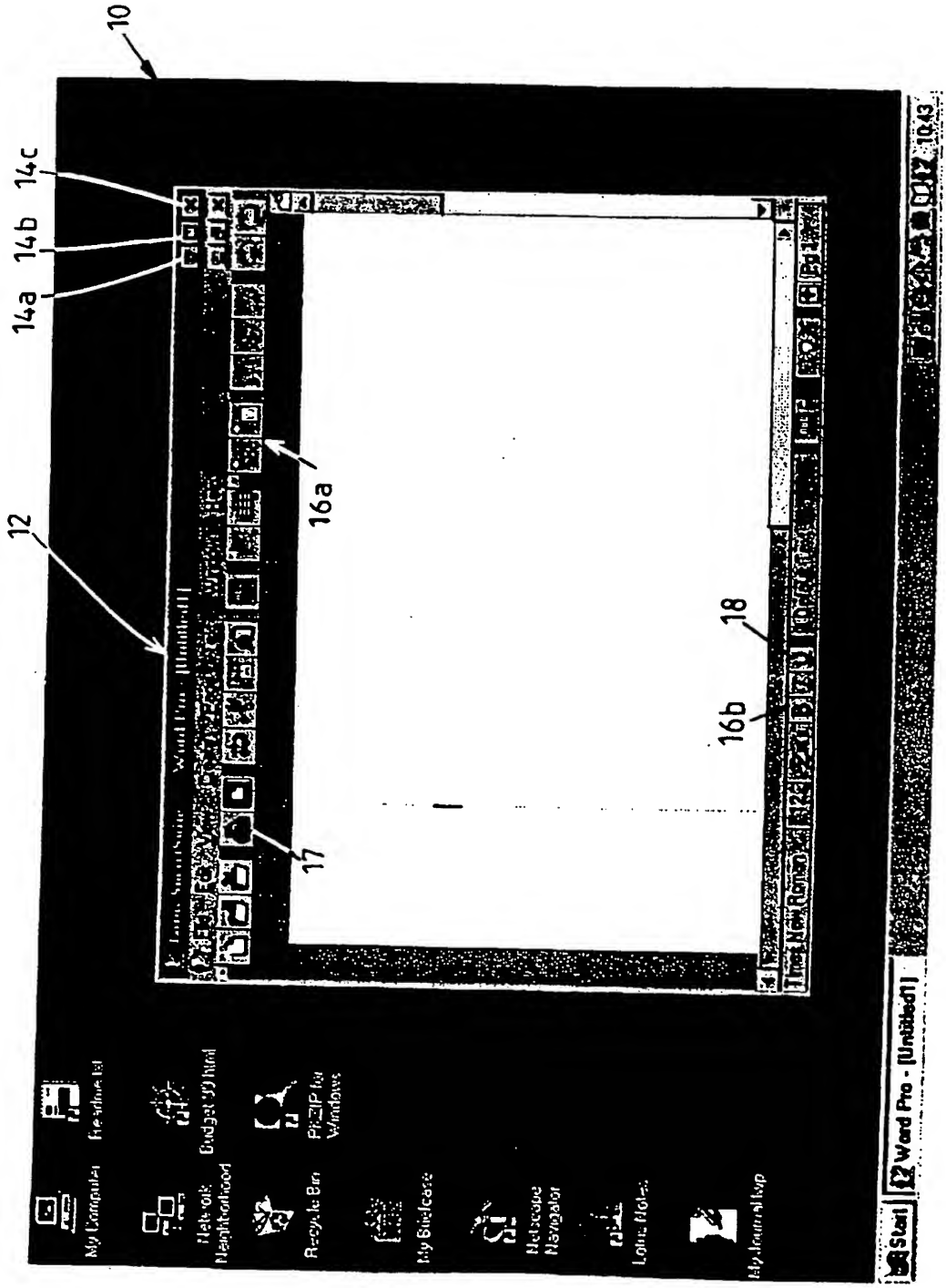


FIG. 2 (PRIOR ART)

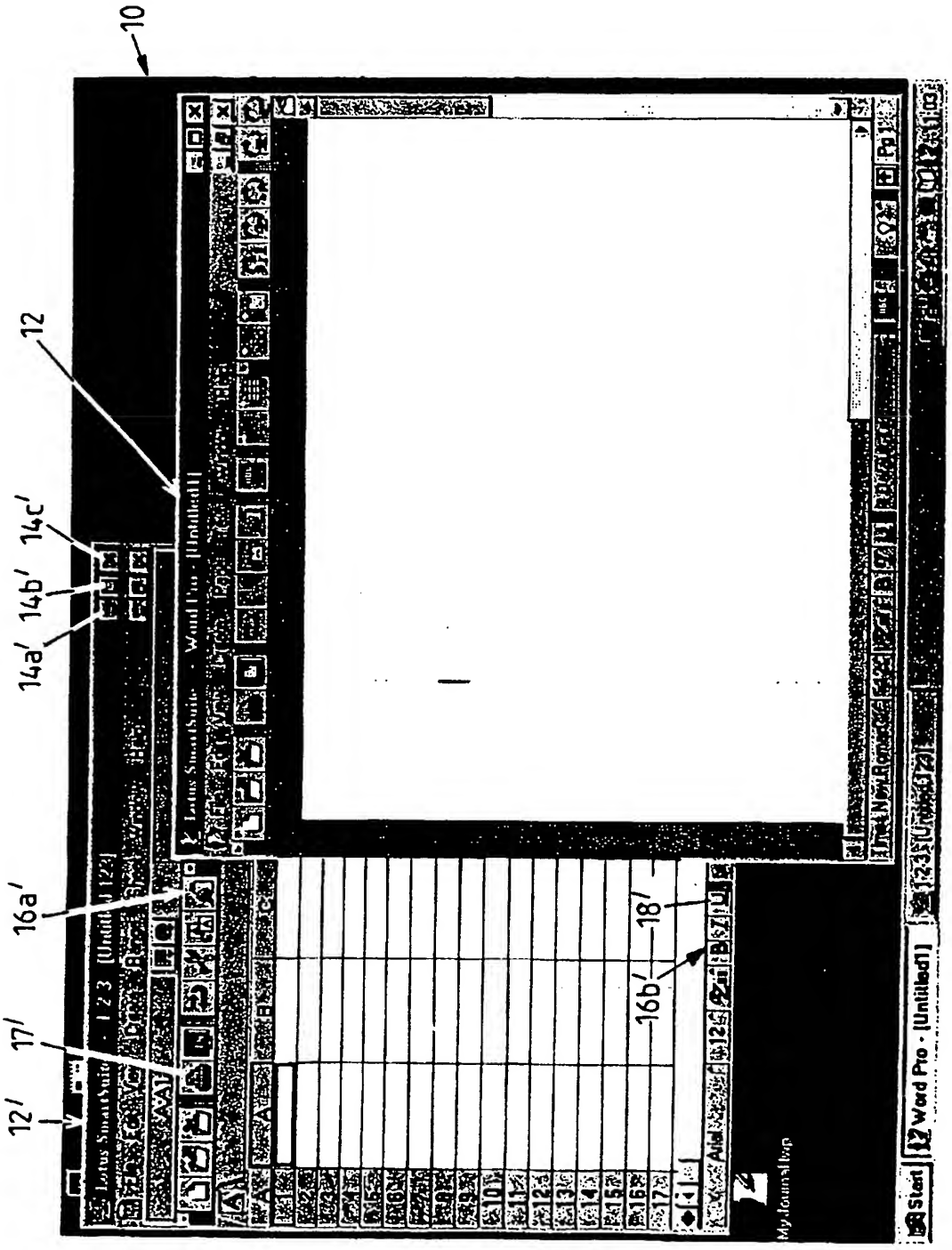


FIG. 3 (PRIOR ART)

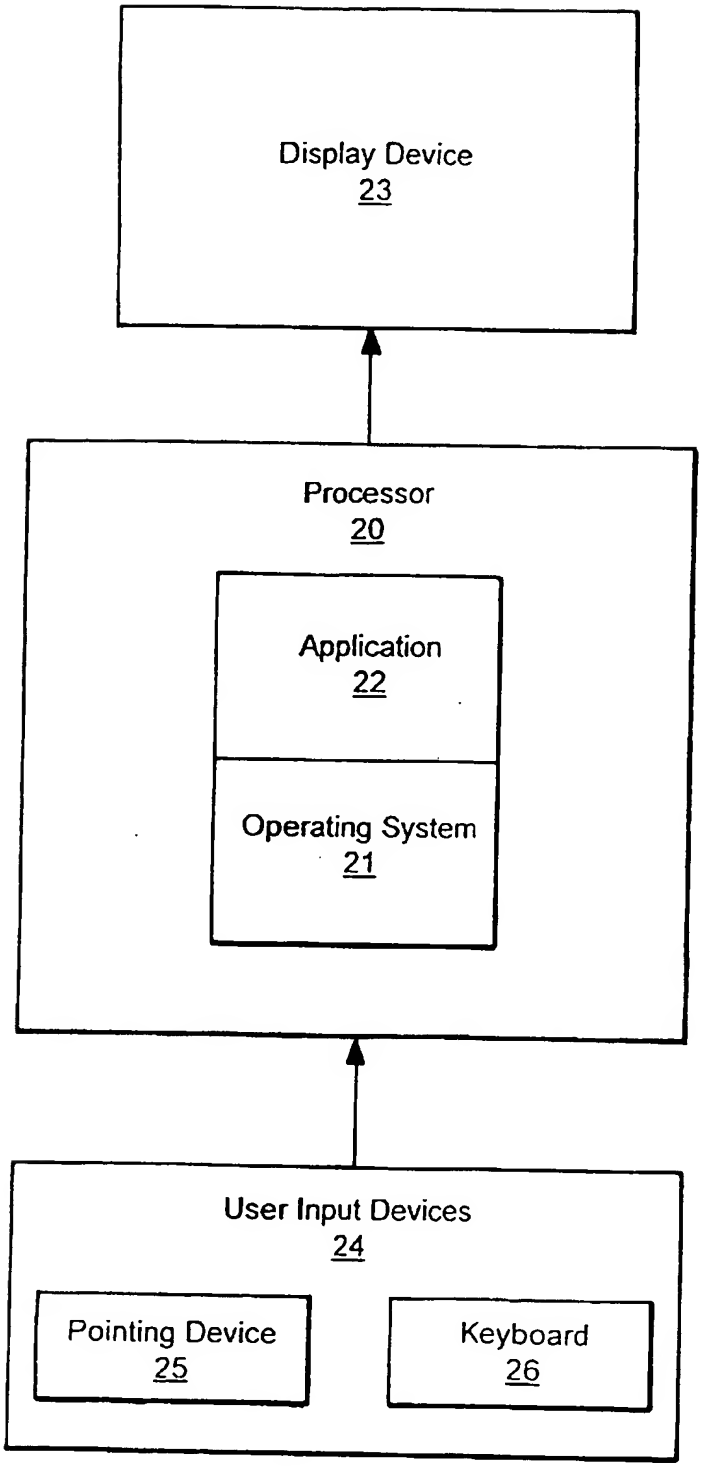


FIG. 4 (PRIOR ART)

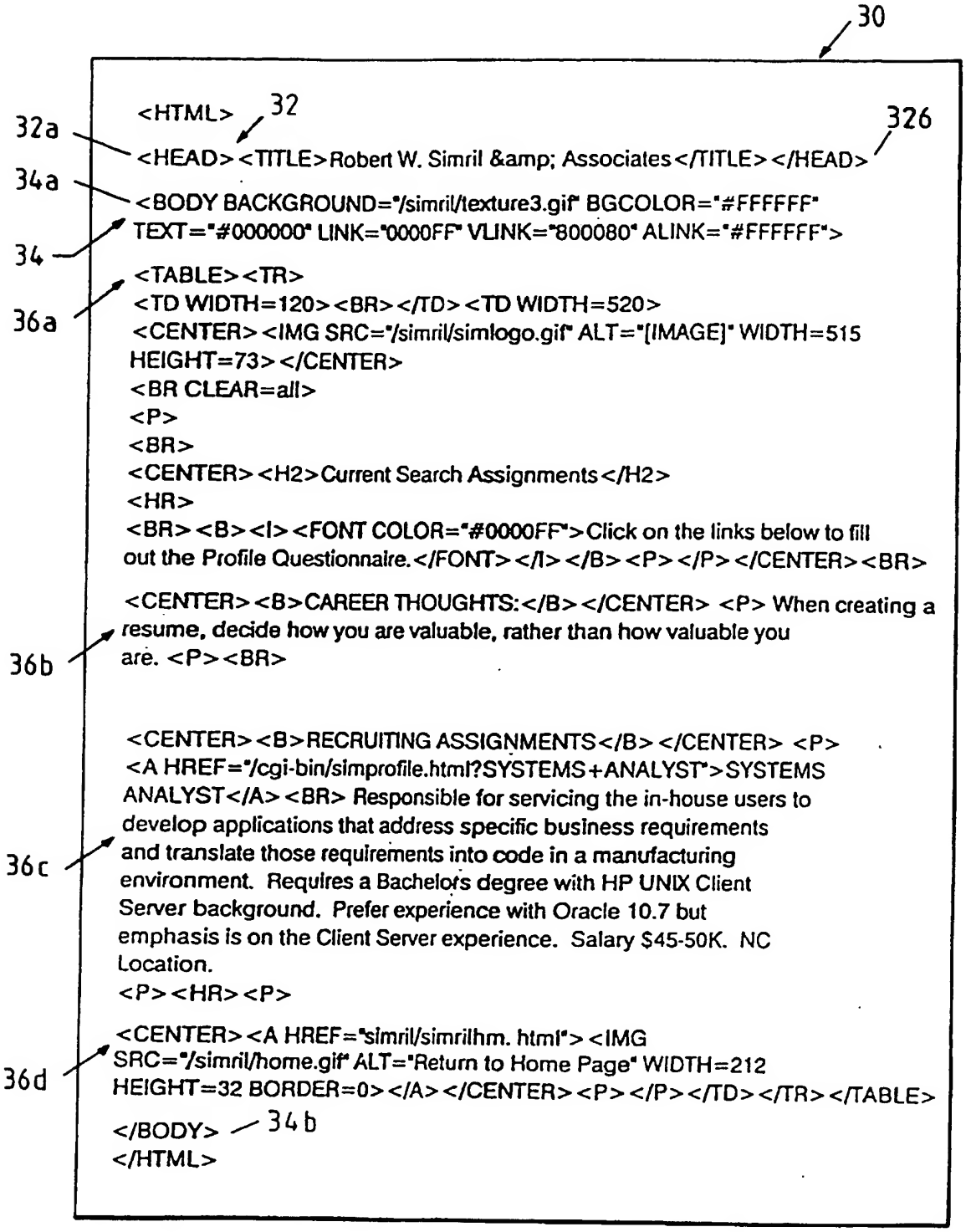


FIG. 5 ( PRIOR ART )

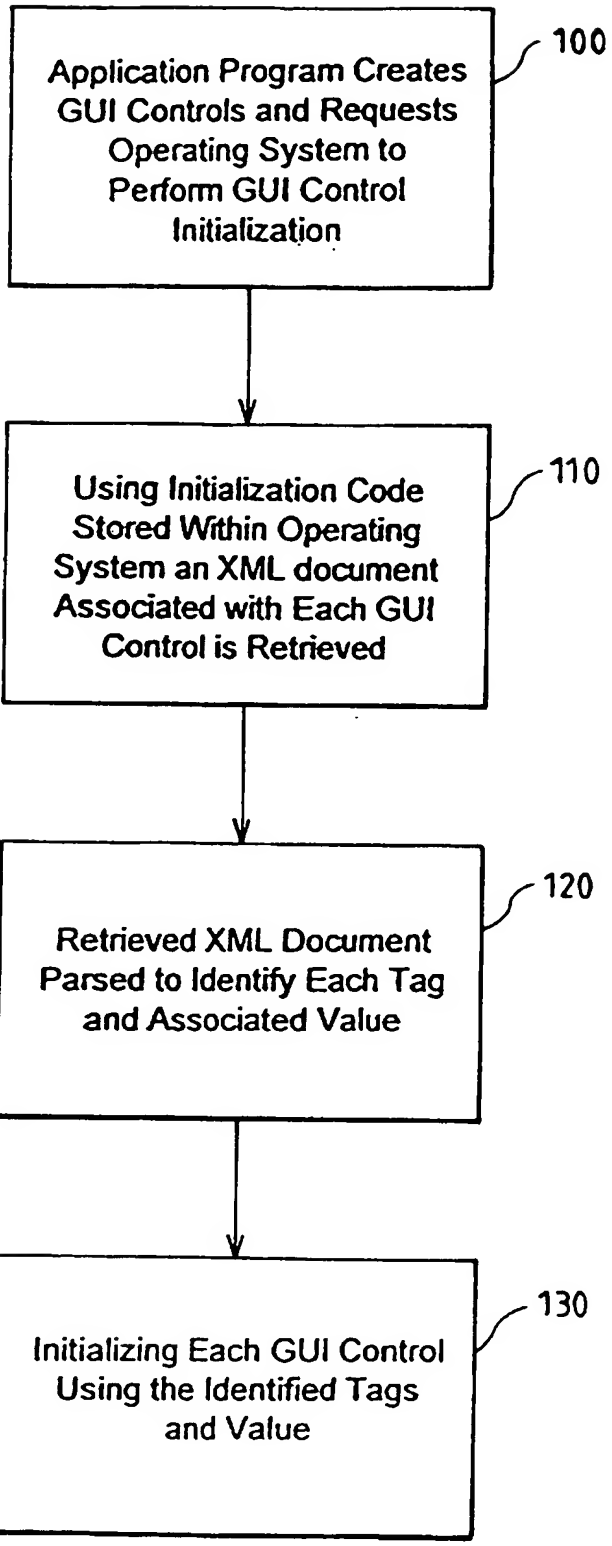
Application Program Creates GUI Controls and Requests Operating System to Perform GUI Control Initialization
100
Using Initialization Code Stored Within Operating System an XML document Associated with Each GUI Control is Retrieved
110
Retrieved XML Document Parsed to Identify Each Tag and Associated Value
120
Initializing Each GUI Control Using the Identified Tags and Value
130

FIG. 6

2350990

# INITIALIZING GRAPHICAL USER INTERFACE CONTROLS WITH EXTENSIBLE MARKUP LANGUAGE-FORMATTED DATA

## Field of the Invention

The present invention relates generally to graphical user interfaces and, more particularly, to graphical user interface controls.

## Background of the Invention

An operating system is a program that, after being initially loaded into a computer by a bootstrap program, manages application programs that perform various data processing tasks. Application programs make use of an operating system by making requests for services through a defined application program interface (API). Examples of application programs include word processors, database programs, web browsers, image editing programs, and communication programs. UNIX, Windows 95, and International Business Machine's (IBM) OS/2, AIX, and OS/390 are all examples of operating systems. (UNIX is a registered trademark licensed exclusively through the Open Group, Windows is a trademark of Microsft Corporation and IBM, OS/2, AIX and OS/390 are trademarks of IBM Corporation.)

The first interactive user interfaces for computers were text-oriented interfaces in which users typed various commands. The command interface of the Microsoft Disk Operating System (DOS) is an example of a text-oriented user interface (Microsoft is a registered trademark of Microsoft Corporation). By contrast, many operating systems today utilize a graphical user interface (GUI) having a "window-like" workspace (often referred to as a "desktop") for displaying application programs, operating system information such as directory information, and program groupings. An exemplary GUI 10 for the Windows 95 operating system is illustrated in Fig. 1. Fig. 2 illustrates a window 12 (referred to as an "application window") displayed within the Windows 95 operating system GUI 10. A word processing application program is associated with the illustrated application window 12.

The building blocks of a GUI are the inventory of display widgets (referred to as "controls") provided therewithin. A GUI control is an element that displays information or provides a specific way for a user to interact with an operating system and/or with an application program. GUI controls may include iconic text buttons, pull-down menus, selection boxes, progress indicators, on-off checkmarks, scroll bars, windows, toggle buttons, forms, and many other devices for displaying information and for inviting, accepting, and responding to user actions within a GUI.

The illustrated application window 12 of Fig. 2 includes GUI controls 14a, 14b for minimizing and maximizing the size of the application window 12, respectively. A GUI control 14c is also provided for closing and removing the application window 12 from the GUI 10. A plurality of GUI controls are also provided within toolbars 16a, 16b, as illustrated. For example, GUI control 17 is provided for printing a document from the application program displayed within the application window 12. GUI control 18 is provided for underlining portions of text within a document in the application program displayed within the application window 12.

Conventionally, an application program displayed within an application window includes code for initializing each GUI control displayed within the application window. The term "initializing" means to give a GUI control its characteristic value(s), such as the location of the GUI control within a GUI and the physical characteristics of the GUI control. For example, in Fig. 2 GUI control 18 is initialized to display the letter "U" on a face thereof in an underlined and bold format of a particular font.

Other application programs launched within a GUI also include various GUI controls for performing various functions. For example, as illustrated in Fig. 3, a second application program window 12' is displayed within the Windows 95 operating system GUI 10. The application window 12' is associated with another application program. The illustrated application window 12' of Fig. 3 includes GUI controls 14a', 14b' for minimizing and maximizing the size of the application window 12', respectively. A GUI control 14c' is also provided for closing the application window 12' and for removing the application window 12' from the GUI 10. A plurality of GUI controls are also provided within toolbars 16a', 16b'. For example, GUI control 17' is provided for printing a document from the application program, and GUI control 18' is provided for underlining portions of text within a document in the application program. As with the application program displayed within the application window 12, the application program displayed within the application window 12' includes code for initializing its own GUI controls within the application window 12.

Conventionally, each respective application program running within a GUI contains and executes code for initializing GUI controls displayed within the GUI, even when an identical GUI control performs an identical function in multiple application programs. As a result, initialization code for GUI controls may be duplicated among application programs. From a software developer's standpoint, it may be desirable to reduce the lines of code within an application program for several reasons. One reason would be to reduce the cost of translating initialization values

within the source code of various application programs into multiple foreign languages. For example, if a GUI control used within multiple application programs by a software developer is configured to display the word "OK", the software developer would need to have the lines of code used to initialize the GUI control in each application program translated into a foreign language equivalent of the word "OK" for each desired foreign language. Another reason would be to reduce the physical size of the code for an application program running on an operating system. Smaller amounts of code for application programs are desirable for pervasive computing devices, such as hand-held devices, personal digital assistants (PDAs), and the like, that may have limited program storage compared with desktop computing devices.

The design of a GUI control is based loosely on object-oriented programming (OOP) techniques. For example, each type of GUI control may be defined as a class (or a subclass under a broad generic GUI control class) and may be associated with a particular window. GUI controls may be initialized with their unique content in a standard way. In a Windows 95 operating system environment, for example, initialization may be performed statically when a GUI control is specified in a resource file (referred to as an "RC file"). For example:

```
BEGIN
            PUSHBUTTON      "&Print",AFX_ID_PREVIEW_PRINT,2,2,44,12
            PUSHBUTTON      "&Next Page",
AFX_ID_PREVIEW_NEXT,48,2,44,12
END
```

In a Java Runtime Environment, for example, GUI controls may be instantiated and initialized dynamically. For example:

```
jButton myButton = new jButton();
myButton.setText("OK");
```

Unfortunately, conventional GUI control initialization techniques, whether static or dynamic, initialize GUI controls with flat strings of text. As a result, initialization of complex GUI controls, such as hierarchical trees, can be difficult using conventional GUI control initialization techniques. This is because conventional GUI control initialization techniques may not have generalized parsing mechanisms associated therewith.

## Disclosure of the Invention

In view of the above, it is therefore an object of the present invention to help reduce the redundancy of GUI control initialization code across multiple application programs running within a GUI.

It is another object of the present invention to help reduce the size of code for application programs running on pervasive computing devices.

It is another object of the present invention to facilitate the initialization of GUI controls using complex data.

These and other objects of the present invention are provided by systems, methods and computer program products for initializing, via a computer operating system, GUI controls for an application program running within the computer operating system GUI. An Extensible Markup Language (XML) document associated with the GUI control is retrieved via initialization code stored within the computer operating system. The XML document may be located locally or remotely with respect to the computer operating system. For example, an XML document may be retrieved from a remotely located computer via a public or private computer network, such as the Internet or an intranet.

An XML document, according to the present invention, includes at least one string of text that indicates how a GUI control is to be displayed within the GUI. The retrieved XML document is parsed by the computer operating system to identify each string of text contained therewithin. The GUI control is then initialized within the GUI using each identified string of text. According to the present invention, each string of text within an XML document contains a value associated with a physical characteristic of the GUI control and at least one pair of opening and closing markup tags surrounding the value. For example, an exemplary string of text contained within an XML document may include the word "OK" (a value) surrounded by the <BOLD> and <ITALICS> markup tags as follows: <BOLD><ITALICS>OK</BOLD></ITALICS>.

The present invention is advantageous because the initialization code for GUI controls can be moved from their respective application programs to the computer operating system within which the application programs are running. This can reduce the amount of code associated with an application program. Furthermore, the need for duplicative initialization code within multiple application programs that utilize the same GUI controls can be eliminated. The present invention is also advantageous because the use of XML-formatted documents can allow complex GUI controls to be initialized using a single call. The use of XML-

formatted documents, according to the present invention, can allow the creation of virtually unlimited types of markup tags that can invoke various advanced functions on GUI controls.

## Brief Description of the Drawings

The invention will now be described, by way of example only, with reference to the accompanying drawings, in which:

Fig. 1 illustrates an exemplary graphical user interface (GUI) for the Windows 95 operating system;

Fig. 2 illustrates an application window displayed within the Windows 95 operating system GUI of Fig. 1;

Fig. 3 illustrates a second application program window displayed within the Windows 95 operating system GUI of Fig. 2;

Fig. 4 illustrates a data processing system in which the present invention may be utilized;

Fig. 5 illustrates an exemplary HTML document; and

Fig. 6 illustrates operations for using an operating system to initialize GUI controls for an application program, according to the present invention.

## Detailed Description of the Invention

As will be appreciated by one of skill in the art, the present invention may be embodied as a method, data processing system, or computer program product. Accordingly, the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment or an embodiment combining software and hardware aspects. Furthermore, the present invention may take the form of a computer program product on a computer-usable storage medium having computer-usable program code means embodied in the medium. Any suitable computer readable medium may be utilized including, but not limited to, hard disks, CD-ROMs, optical storage devices, or magnetic storage devices.

Computer program code for carrying out operations of the present invention is preferably written in an object oriented programming language such as Java, Smalltalk or C++ (Java is a trademark of Sun Microsystems Inc.). Java is an object-oriented programming language developed by Sun Microsystems, Mountain View, California. Java is a portable and architecturally neutral language. Java source code is

compiled into a machine-independent format that can be run on any machine with a Java runtime system known as the Java Virtual Machine (JVM). The JVM is defined as an imaginary machine that is implemented by emulating a processor through the use of software on a real machine. Accordingly machines running under diverse operating systems, including UNIX, Windows 95, Windows NT, and MacIntosh having a JVM can execute the same Java program (MacIntosh is a trademark of Apple Corporation).

However, the computer program code for carrying out operations of the present invention may also be written in conventional procedural programming languages, such as the "C" programming language. The program code may execute entirely on the user's computer, as a stand-alone software package, or it may execute partly on the user's computer and partly on a remote computer. In the latter scenario, the remote computer may be connected to the user's computer through a LAN or a WAN, or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

Fig. 4 illustrates a data processing system in which the present invention may be utilized. As seen in Fig. 4, a data processor 20 may have an operating system, 21 resident therein. An application program 22 may be running on the operating system 21. The processor 20 displays information on a display device 23. The display device 23 has a plurality of picture elements (collectively referred to as a screen) which may define the appearance of a GUI displayed on the display device 23. The contents of the screen 23 and, therefore, the appearance of the GUI, may be controlled or altered by the application program 22 or the operating system 21, either individually or in combination. For obtaining input from a user, the operating system 21, the application program 22, or both, may utilize user input devices 24. User input devices 24 may include a pointing device 25 and a keyboard 26 or other input devices known to those of skill in the art.

## HTML

As is known to those skilled in this art, a web page is conventionally formatted via a standard page description language such as HyperText Markup Language (HTML), and typically displays text and graphics, and can play sound, animation, and video data. HTML provides basic document formatting and allows a web page developer to specify hypertext links (typically manifested as highlighted text) to other servers and files. When a user selects a particular hypertext link, a web browser reads and interprets the address, called a URL (Uniform Resource Locator) associated with the link, connects the web browser with the web server at that address, and makes an HTTP request for the web page identified in the link. The web server then sends the requested web

page to the client in HTML format which the browser interprets and displays to the user.

An HTML document can be comprised of text, images and a variety of objects, each of which are surrounded by various markup language tags that control format attributes and identify different portions of the document (*i.e.*, <tag_name>text</tag_name>). HTML documents are typically written and stored in ASCII text format using a text editor.

Markup language tags are conventionally denoted by the "< >" symbols, with the actual tag between the brackets. Most markup language tags have an opening tag "<tag>" and a closing tag "</tag>." For example, to make a line of text appear as a heading, the following tags enclose the text: <H3></H3>. (<H3>This text appears as a heading.</H3>). An exemplary HTML document 30 is illustrated in Fig. 5. The illustrated HTML document 30 includes a header section 32 demarcated by <HEAD> tags 32a, 32b. A body section 34, demarcated by <BODY> tags 34a, 34b, includes various "content" portions 36a, 36b, 36c, 36d. It is these content portions 36a, 36b, 36c, 36d that contain information displayed to a user viewing the HTML document 30 with a web browser.

The HTML language is described in the HTML Reference Manual, January 2, 1996, published by Sandia National Laboratories, which is incorporated herein by reference, in its entirety. Because markup language tags are used only within an HTML source document, the markup language tags do not appear within the actual displayed web page. It is to be understood that the terms "tag" and "markup language tag" can be used interchangeably.

## Extensible Markup Language

Extensible Markup Language (XML) is currently a formal recommendation from the World Wide Web Consortium as a way to make the Web a more versatile tool. XML is similar to HTML in that both languages utilize tags to describe the contents of a page or file. HTML, however, describes the content of a Web page (mainly text and graphic images) only in terms of how content is to be displayed and interacted with. XML describes the content in terms of what data is being described. For example, a <PHONENUM> tag could indicate that the data following the tag is a phone number. This means that an XML file can be processed purely as data by a program or it can be stored with similar data on another computer or, like an HTML file, that it can be displayed. For example, depending on how the application program in the receiving computer wanted to handle the phone number, it could be stored, displayed, or dialed. XML is "extensible" because, unlike HTML, XML markup tags can be unlimited and can be self-defining.

Initializing GUI Controls With XML Data

According to the present invention, GUI controls used within an application program are initialized by code residing within a computer operating system on which the application program is running. The computer operating system calls an XML document associated with each GUI control and initializes each GUI control using parameters passed in the associated XML document. The XML document may be a locally stored document or a remotely stored document. In the case where an XML document is remotely stored, an operating system calling the XML document may identify the XML document via a URL. For example:

```
Button myOKButton = new Button();    // Create the button
myOKButton.XMLInitialize("http://uirepos.ibm.com/strings#OKButton"); //
Initialize button
```

The present invention will now be described with reference to flowchart illustrations (Fig. 6) of methods, apparatus (systems) and computer program products according to an embodiment of the invention. It will be understood that each block of the flowchart illustrations of Fig. 6 and combinations of blocks in the flowchart illustrations of Fig. 6, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions specified in the flowchart block or blocks.

The computer program instructions may be executed by a processor to cause a series of operational steps to be performed by the processor to produce a computer implemented process such that the instructions which execute on the processor provide steps for implementing the functions specified in the flowchart block or blocks. The computer program instructions may also be stored in a computer-readable memory that can direct a computer or other programmable data processing apparatus to function in a particular manner, such that the instructions stored in the computer-readable memory produce an article of manufacture including instruction means which implement the function specified in the flowchart block or blocks.

Referring now to Fig. 6, operations, according to the present invention, for using a computer operating system to initialize GUI controls for an application program running within the computer operating system GUI are schematically illustrated. To initialize and display GUI controls associated with the application program, the application program

creates GUI controls and then requests the operating system to perform initialization of the created GUI controls (Block 100). Using initialization code stored within the computer operating system, the computer operating system retrieves an XML document associated with each respective GUI control (Block 110). The operating system parses each retrieved XML document and passes each tag and associated value to the GUI control associated with a respective XML document (Block 120). Each GUI control then initializes itself using the passed tags and values and is displayed within the GUI (Block 130).

For example, an application program utilizes a button (*i.e.*, a GUI control) labeled "OK" that is normally displayed within an application window associated with the application program. The application program creates the button and then requests the computer operating system to perform initialization of the button (Block 100). Using the following exemplary initialization code, the computer operating system retrieves an XML document associated with the OK button (Block 110):

```
okButton.setText(<SYSTEM-REPOSITORY ID="OK-BUTTON-TEXT"/>);.
```

The associated XML document contains pairs of opening and closing tags that are configured to display the word "OK" in italics and bold (*i.e.*, <BOLD><ITALICS> <VALUE>OK</VALUE> </ITALICS></BOLD>). The computer operating system parses the associated XML document and passes each tag (*i.e.*, <BOLD></BOLD> and <ITALICS></ITALICS>) and associated value (*i.e.*, "OK") to the button (Block 120). The button initializes itself using the passed tags and value and is displayed within the application window of the GUI (Block 130).

The use of an XML document in the initialization of GUI controls, according to the present invention, can allow for the creation of virtually unlimited types of markup tags that can invoke various advanced functions on a GUI control. For example, a markup tag such as "<WINDOW-TEXT [VALUE-URI="uri-spec"]>" can be used to set the text value of a GUI control. A direct value (*i.e.*, the word "OK") may be included in the tag, or the tag may reference data remotely located within a computer network (*e.g.*, the Internet) using a Uniform Resource Identifier (URI) reference. As is known by those skilled in the art, URIs, which are a subset of URLs, allow a user to address a document and even a specific section of a document.

The following initialization code from an XML document illustrates initializing GUI controls according to the present invention.

```
<LIST>
  <ITEM>Alabama</ITEM>
  <ITEM>Pennsylvania</ITEM>
  <ITEM SELECTED="TRUE">North Carolina</ITEM>
</LIST>
```

The markup tag <LIST> is used to initialize a GUI control with a list. The above initialization code, provided via an XML document, performs a complex function heretofore unavailable using conventional initialization techniques. First, the respective text lines of a "list" GUI control are initialized. Second, the default selection state within the list is set (*i.e.*, North Carolina).

Another example of initializing GUI controls with XML documents according to the present invention is illustrated below. The markup tag <EDIT> is used to initialize a GUI control with the ability to edit multiple lines of text, as follows:

```
<EDIT>
  <LINE>Hello, My name is<BOLD>John B. Smith</BOLD></LINE>
  <LINE>I live in Raleigh, <ITALICS>North</ITALICS> Carolina</LINE>
  <LINE></LINE>
  <LINE>Bye Now</LINE>
</EDIT>
```

In addition, the present invention may be utilized to perform complex GUI control initializations such as hierarchical trees. An exemplary hierarchical tree is created conventionally in Java as follows:

```
Jtree StateCityTree = new Jtree();                    // Create root of tree
TreeNode state1Node = new TreeNode("Alabama");    // Create children of
the tree
TreeNode city1Node1 = new TreeNode("Mobile");
TreeNode city1Node2 = new TreeNode("Birmingham");
TreeNode state2Node = new TreeNode("Arizona");
TreeNode city2Node1 = new TreeNode("Phoenix");
TreeNode city2Node2 = new TreeNode("Flagstaff");
state1Node.addNode( city1Node1 );
state1Node.addNode( city1Node2 );
state2Node.addNode( city2Node1 );
state2Node.addNode( city2Node2 );
StateCityTree.addNode( state1Node );
StateCityTree.addNode( state2Node );
```

Thus, each application program displaying the above hierarchical tree GUI control would contain the above code for initializing the hierarchical tree.

According to the present invention, the hierarchical tree can be initialized using an XML document, either locally or remotely located, as follows:

```
Jtree StateCityTree = new Jtree();          // Create root of tree
StateCityTree.XMLInitialize( treeInitURL );
```

Somewhere in a computer network, such as the Internet, or in a data area of the application program displaying the hierarchical tree is the requested XML document "StateCityTree."

```
<TREE>
  <CHILD>Alabama
    <CHILD>Mobile</CHILD>
    <CHILD>Birmingham</CHILD>
  </CHILD>
  <CHILD>Arizona
    <CHILD>Phoenix</CHILD>
    <CHILD>Flagstaff</CHILD>
  </CHILD>
</TREE>
```

When more nodes are added to the tree, the XML document size increases, but no additional code lines are needed.

## CLAIMS

1. A method of initializing a graphical user interface (GUI) control for an application program running within a computer operating system GUI, the method comprising the steps of:

retrieving an Extensible Markup Language (XML) document associated with the GUI control via initialization code stored within the computer operating system, wherein the identified XML document includes at least one string of text that indicates how the GUI control is to be displayed within the GUI;

parsing the retrieved XML document to identify the at least one string of text; and

initializing the GUI control within the GUI using the identified at least one string of text in the retrieved XML document.

2. A method according to Claim 1 wherein the step of retrieving an XML document associated with the GUI control comprises executing the GUI control initialization code within the computer operating system.

3. A method according to Claim 1 wherein the step of parsing the retrieved XML document is performed by the computer operating system.

4. A method according to Claim 1 wherein the at least one string of text included within the XML document comprises:

at least one value associated with a physical characteristic of the GUI control; and

at least one pair of opening and closing markup tags surrounding the at least one value.

5. A method according to Claim 4 wherein the at least one value comprises a plurality of values.

6. A method according to Claim 4 wherein the at least one pair of opening and closing markup tags comprises a plurality of pairs of markup tags.

7. A method according to Claim 1 wherein the step of retrieving an XML document is performed in response to a request from the application program to the computer operating system to initialize the GUI control.

8. A method according to Claim 1 wherein the step of retrieving an XML document comprises retrieving the XML document from a remotely located computer via a computer network.

9. A method of initializing a graphical user interface (GUI) control for an application program running within a computer operating system GUI, the method comprising the steps of:

retrieving an Extensible Markup Language (XML) document associated with the GUI control via initialization code stored within the computer operating system, wherein the identified XML document includes a value associated with a physical characteristic of the GUI control and a pair of opening and closing markup tags surrounding the value that indicates how the physical characteristic of the GUI control is to be displayed within the GUI;

parsing the retrieved XML document within the computer operating system to identify the value and the pair of opening and closing markup tags; and

initializing the GUI control within the GUI using the identified value and the pair of opening and closing markup tags in the retrieved XML document.

10. A method according to Claim 9 wherein the step of retrieving an XML document associated with the GUI control comprises executing the GUI control initialization code within the computer operating system.

11. A method according to Claim 9 wherein the step of retrieving an XML document is performed in response to a request from the application program to the computer operating system to initialize the GUI control.

12. A method according to Claim 9 wherein the step of retrieving an XML document comprises retrieving the XML document from a remotely located computer via a computer network.

13. A system for initializing a graphical user interface (GUI) control for an application program running within a computer operating system GUI, comprising:

means for retrieving an Extensible Markup Language (XML) document associated with the GUI control via initialization code stored within the computer operating system, wherein the identified XML document includes at least one string of text that indicates how the GUI control is to be displayed within the GUI;

means for parsing the retrieved XML document to identify the at least one string of text; and

means for initializing the GUI control within the GUI using the identified at least one string of text in the retrieved XML document.

14. A system according to Claim 13 wherein the means for retrieving an XML document associated with the GUI control comprises means for executing the GUI control initialization code within the computer operating system.

15. A system according to Claim 13 wherein the means for parsing the retrieved XML document is included within the computer operating system.

16. A system according to Claim 13 wherein the at least one string of text included within the XML document comprises:

at least one value associated with a physical characteristic of the GUI control; and

at least one pair of opening and closing markup tags surrounding the at least one value.

17. A system according to Claim 16 wherein the at least one value comprises a plurality of values.

18. A system according to Claim 16 wherein the at least one pair of opening and closing markup tags comprises a plurality of pairs of opening and closing markup tags.

19. A system according to Claim 13 wherein the means for retrieving an XML document is responsive to receiving a request from the application program to initialize the GUI control.

20. A system according to Claim 13 wherein the means for retrieving an XML document comprises means for retrieving the XML document from a remotely located computer via a computer network.

21. A system for initializing a graphical user interface (GUI) control for an application program running within a computer operating system GUI, comprising:

means for retrieving an Extensible Markup Language (XML) document associated with the GUI control via initialization code stored within the computer operating system, wherein the identified XML document includes a value associated with a physical characteristic of the GUI control and a pair of opening and closing markup tags surrounding the value that indicates how the physical characteristic of the GUI control is to be displayed within the GUI;

means for parsing the retrieved XML document within the computer operating system to identify the value and the pair of opening and closing markup tags; and

means for initializing the GUI control within the GUI using the identified value and pair of opening and closing markup tags in the retrieved XML document.

22. A system according to Claim 21 wherein the means for retrieving an XML document associated with the GUI control comprises means for executing the GUI control initialization code within the computer operating system.

23. A system according to Claim 21 wherein the means for retrieving an XML document is responsive to receiving a request from the application program to initialize the GUI control.

24. A system according to Claim 21 wherein the means for retrieving an XML document comprises means for retrieving the XML document from a remotely located computer via a computer network.

25. A computer program product for initializing a graphical user interface (GUI) control for an application program running within a computer operating system GUI, the computer program product comprising a computer usable storage medium having computer readable program code means embodied in the medium, the computer readable program code means comprising:

computer readable program code means for retrieving an Extensible Markup Language (XML) document associated with the GUI control via initialization code stored within the computer operating system, wherein the identified XML document includes at least one string of text that indicates how the GUI control is to be displayed within the GUI;

computer readable program code means for parsing the retrieved XML document to identify the at least one string of text; and

computer readable program code means for initializing the GUI control within the GUI using the identified at least one string of text in the retrieved XML document.

26. A computer program product according to Claim 25 wherein the computer readable program code means for retrieving an XML document associated with the GUI control comprises computer readable program code means for executing the GUI control initialization code within the computer operating system.

27. A computer program product according to Claim 25 wherein the computer readable program code means for parsing the retrieved XML document is included within the computer operating system.

28. A computer program product according to Claim 25 wherein the at least one string of text included within the XML document comprises:

at least one value associated with a physical characteristic of the GUI control; and

at least one pair of opening and closing markup tags surrounding the at least one value.

29. A computer program product according to Claim 28 wherein the at least one value comprises a plurality of values.

30. A computer program product according to Claim 28 wherein the at least one pair of opening and closing markup tags comprises a plurality of pairs of opening and closing markup tags.

31. A computer program product according to Claim 25 wherein the computer readable program code means for retrieving an XML document is responsive to receiving a request from the application program to initialize the GUI control.

32. A computer program product according to Claim 25 wherein the computer readable program code means for retrieving an XML document comprises computer readable program code means for retrieving the XML document from a remotely located computer via a computer network.

33. A computer program product for initializing a graphical user interface (GUI) control for an application program running within a computer operating system GUI, the computer program product comprising a computer usable storage medium having computer readable program code means embodied in the medium, the computer readable program code means comprising:

computer readable program code means for retrieving an Extensible Markup Language (XML) document associated with the GUI control via initialization code stored within the computer operating system, wherein the identified XML document includes a value associated with a physical characteristic of the GUI control and a pair of opening and closing markup tags surrounding the value that indicates how the physical characteristic of the GUI control is to be displayed within the GUI;

computer readable program code means for parsing the retrieved XML document within the computer operating system to identify the value and the pair of opening and closing markup tags; and

computer readable program code means for initializing the GUI control within the GUI using the identified value and the pair of opening and closing markup tags in the retrieved XML document.

34. A computer program product according to Claim 33 wherein the computer readable program code means for retrieving an XML document associated with the GUI control comprises computer readable program code means for executing the GUI control initialization code within the computer operating system.

35. A computer program product according to Claim 33 wherein the computer readable program code means for retrieving an XML document is responsive to receiving a request from the application program to initialize the GUI control.

36. A computer program product according to Claim 33 wherein the computer readable program code means for retrieving an XML document comprises computer readable program code means for retrieving the XML document from a remotely located computer via a computer network.

CHARTER MARK

The Patent Office
PATENTS • DESIGNS • COPYRIGHT • TRADE MARKS

INVESTOR IN PEOPLE

18

| | | | |
|---|---|---|---|
| **Application No:** | GB 9925012.8 | **Examiner:** | Joe McCann |
| **Claims searched:** | All | **Date of search:** | 3 October 2000 |

## Patents Act 1977
## Search Report under Section 17

**Databases searched:**

UK Patent Office collections, including GB, EP, WO & US patent specifications, in:

UK Cl (Ed.R): H4T(TDXX)

Int Cl (Ed.7): G06F(9/44,9/45)

Other: Online: WPI, INSPEC, EPODOC, JAPIO, TDB, COMPUTER, Internet

**Documents considered to be relevant:**

| Category | Identity of document and relevant passage | Relevant to claims |
|---|---|---|
| XP | WO 99/17193A1 (UNISYS CORP) - see whole document | 1-36 |
| Y | WO 98/06033A1 (AGRANAT SYSTEMS INC) - see abstract | 1-36 |
| Y | "XML Update: An XML GUI Programming Tool", William Gee, Oct 1, 1998, www.webtools.com/...intableArticle?doc_id=TLS19980805S0008 | 1-36 |

| | | | |
|---|---|---|---|
| X | Document indicating lack of novelty or inventive step | A | Document indicating technological background and/or state of the art. |
| Y | Document indicating lack of inventive step if combined with one or more other documents of same category. | P | Document published on or after the declared priority date but before the filing date of this invention. |
| & | Member of the same patent family | E | Patent document published on or after, but with priority date earlier than, the filing date of this application. |